COMPUTER SCIENCE PROJECT

On

RB Airlines Reservation System

Efforts By:

Radhika Bansal
Roll No:37 (XII – A)
2020 – 2021

ACKNOWLEDGEMENT

I, Radhika Bansal, student of class XII-A of Apeejay School Pitampura, am grateful to our **Principal** for providing me with an opportunity and resources for carrying forward my project on the topic:

"RB Airlines Reservation System"

I am thankful to my computer science teacher **Ms. Puja Malhotra**. Her valuable guidance, support and supervision all through this project were very helpful.

The progress in **science and technology** has been a great help to me for finding the information as well as making my project effective.

I am also thankful to my **parents** for their constant motivation and support and my classmates, whose suggestions were appreciable.

I am thus, obliged and thankful to all those who have directly or indirectly helped me to complete this project and guide me through in whatever way they could.

CERTIFICATE

This is hereby to certify that the following project work in the subject Computer Science has been carried out with complete sincerity and satisfaction by **Radhika Bansal of Class XII - A, Apeejay School, Pitampura.** The project is titled:

"RB Airlines Reservation System"

Teacher's Signature

INDEX

CONTENTS **PAGE No.**

1. Introduction……1

2. Source Code……..3

3. Output Screens……43

4. Enhancements……..53

5. Hardware/Software Required…….54

6. Analysis……55

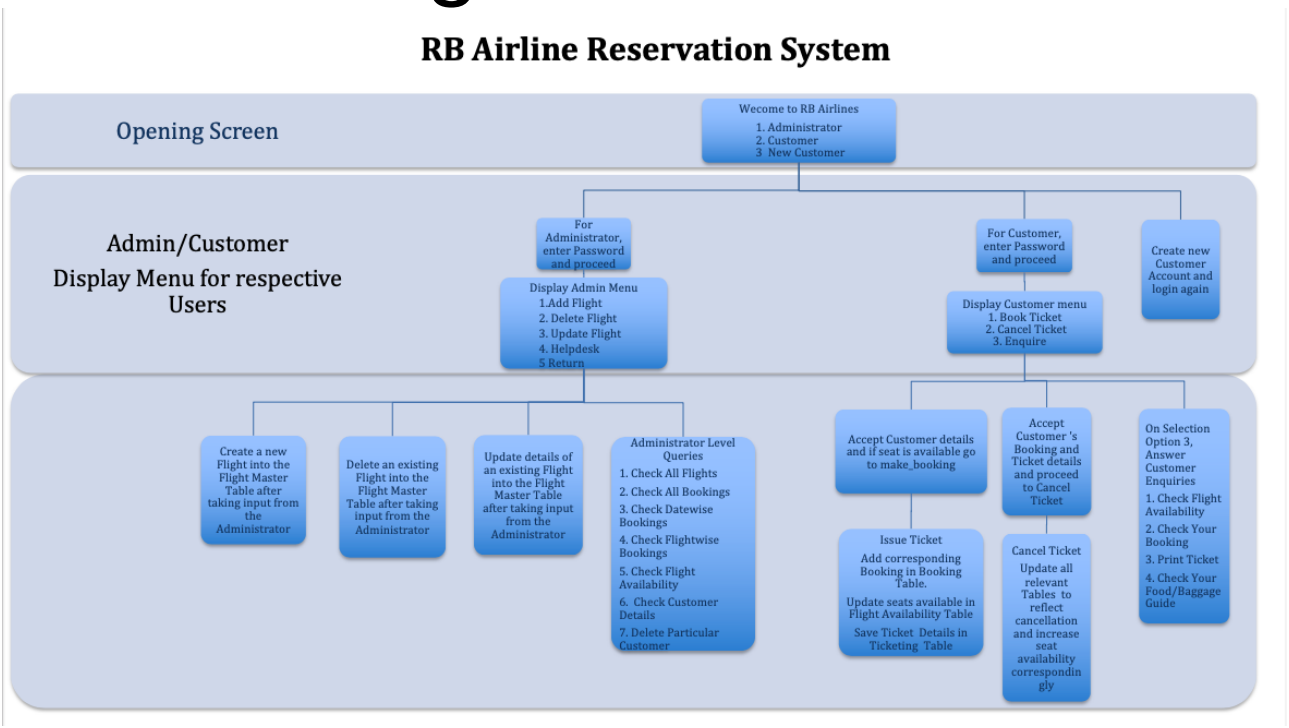
7. Bibliography……..……56

INTRODUCTION

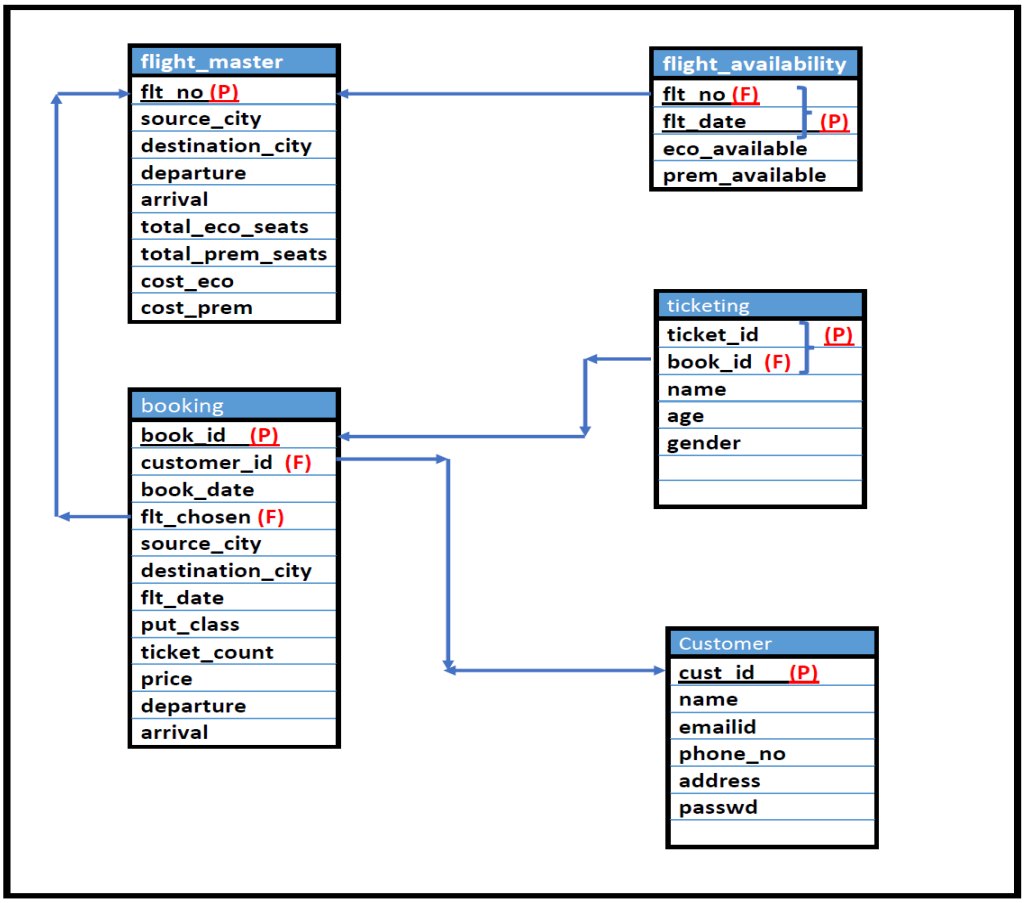
Online Reservation and Booking Systems have always fascinated me. Thus, when I got the opportunity to choose a topic for my Computer Science project, I decided to make my project on an airline reservation system and named it “RB Airlines Reservation System”.

In this project, the focus is on booking tickets for RB Airlines on various routes in the domestic market. Some of the actual routes between metropolitan cities of India have been used. The System is easy to use if one keeps on following the steps as mentioned at the time of execution i.e. the program is user friendly. The system coded in Python uses MySQL Python Connector to maintain a MySQL database (shown on next page) containing data regarding flights, customers and their bookings and focuses on making bookings, printing tickets, cancellation of tickets and all enquires related to flight availability and booking information. The System functionality has been broadly depicted by the block diagram on the next page. In the end, I would say that I have given my sincere efforts to this project.

Block Diagram and Database



MySQL RB_AIRLINES Database Structure with Primary Key / Foreign Key Relationships



SOURCE CODE

There are two code files:

1) **Create DB file**-
   This has the python interfacing with SQL code that creates the database and its tables on the system.

2) **Main source code-**
   The main source code for the program

CREATE DB FILE

```
import mysql.connector
mydb=mysql.connector.connect(host="localhost",user="root",pas
sword="root",database="RB_AIRLINES")
if mydb.is_connected():
   print("Ok....Connection Made....")
   print(mydb)


mycursor=mydb.cursor()
mycursor.execute("create database RB_AIRLINES")
mycursor=mydb.cursor()
mycursor.execute("show databases")
for x in mycursor:
   print(x)


mycursor.execute("create table flight_master (flt_no char(4)
PRIMARY KEY, source_city varchar(20) NOT NULL, destination_city
varchar(20) NOT NULL, departure char(10) NOT NULL, arrival
char(10) NOT NULL, total_eco_seats integer, total_prem_seats
integer, cost_eco float, cost_prem float)")


mycursor.execute("create table flight_availability (flt_no char(4),
flt_date date, eco_available integer, prem_available integer,
PRIMARY KEY(flt_no,flt_date), FOREIGN KEY(flt_no) REFERENCES
flight_master(flt_no))")


mycursor.execute("create table customer(cust_id varchar(10) NOT
NULL PRIMARY KEY, name varchar(30) NOT NULL, emailid
varchar(20), phone_no char(10), address varchar(30), passwd
varchar(8))")
```

```
mycursor.execute("create table booking(book_id int PRIMARY
KEY, customer_id varchar(10), book_date date,flt_chosen char(4)
NOT NULL, source_city varchar(20) NOT NULL, destination_city
varchar(20) NOT NULL,flt_date date NOT NULL, put_class
varchar(20) NOT NULL, tickets_count int NOT NULL, price float,
departure char(10) NOT NULL, arrival char(10) NOT NULL,
FOREIGN KEY(flt_chosen) REFERENCES
flight_master(flt_no),FOREIGN KEY(customer_id) REFERENCES
customer(cust_id) )")


mycursor.execute("create table ticketing(ticket_id int NOT NULL,
book_id int NOT NULL, name varchar(30) NOT NULL, age int NOT
NULL, gender char(1) check (gender='M' or gender='F'), PRIMARY
KEY(ticket_id,book_id), FOREIGN KEY (book_id) REFERENCES
booking(book_id))")


mydb.commit()
mycursor.close()
mydb.close()
```

MAIN SOURCE CODE

```
#RB AIRLINE SYSTEM
from os import system, name
from time import sleep
import json
import datetime
import getpass
import mysql.connector
import random
from mysql.connector import Error
import time
#import twilio


# defining our clear function to clear screen
def clear():
   if name == 'nt': #nt stands for windows where you clear screen
with cls)
      _ = system('cls')

   else:
      _ = system('clear')

def add_flight():
   print("ADD FLIGHT".center(100))
   print("--------------------".center(100))
   print("Enter New Flight Details:  ")
   print()
   f_no=input("Enter Flight Number ")
   s_city=input("Enter Source City ")
   des_city=input("Enter Destination City ")
   dep=input("Enter Departure Time (for eg 1200 hours )")
   arr=input("Enter Arrival Time (for eg 1200 hours) ")
   tot_eco=int(input("Enter Total Number of Economy Seats in
Flight "))
```

```
    tot_prem=int(input("Enter Total Number of Premium Seats in
Flight "))
    cost_eco=float(input("Enter  Economy Seat Price "))
    cost_prem=float(input("Enter  Premium Seat Price "))
    print()
    try:
        mySql_insert_query = """INSERT INTO flight_master (flt_no,
source_city, destination_city, departure, arrival, total_eco_seats,
total_prem_seats, cost_eco, cost_prem)
          VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s) """
        recordTuple = (f_no, s_city, des_city, dep, arr, tot_eco,
tot_prem, cost_eco, cost_prem)
        mycursor = mydb.cursor()
        mycursor.execute(mySql_insert_query, recordTuple)

        mySql_insert_query = """INSERT INTO flight_availability
(flt_no, flt_date, eco_available, prem_available) VALUES
(%s,%s,%s,%s) """
        avail_dt= '2020-03-28'
        recordTuple = (f_no,avail_dt ,tot_eco,tot_prem)
        mycursor = mydb.cursor()
        mycursor.execute(mySql_insert_query, recordTuple)

        mySql_insert_query = """INSERT INTO flight_availability
(flt_no, flt_date, eco_available, prem_available) VALUES
(%s,%s,%s,%s) """
        avail_dt= '2020-03-29'
        recordTuple = (f_no,avail_dt ,tot_eco,tot_prem)
        mycursor = mydb.cursor()
        mycursor.execute(mySql_insert_query, recordTuple)

        mySql_insert_query = """INSERT INTO flight_availability
(flt_no, flt_date, eco_available, prem_available) VALUES
(%s,%s,%s,%s) """
        avail_dt= '2020-03-30'
        recordTuple = (f_no,avail_dt ,tot_eco,tot_prem)
        mycursor = mydb.cursor()
        mycursor.execute(mySql_insert_query, recordTuple)
```

```
        mydb.commit()
        print(mycursor.rowcount, "Record inserted successfully into
Flight Master table")
        print()

    except mysql.connector.Error as error:
        print("Failed to insert record into table {}".format(error))
        print()

    finally:
        print()
        mycursor.close()
        ##if (mydb.is_connected()):
              ##connection.close()
        ##print("MySQL connection is closed")

def del_flight():
    print("DELETE FLIGHT".center(100))
    print("----------------------".center(100))
    print()
    f_no=input("Enter Flight Number of Flight to be Deleted ")
    print()
    try:
        mycursor = mydb.cursor()
        print("Displaying Flight Details Before Deletion ")
        sql_select_query = """select * from flight_master where flt_no
= %s"""
        mycursor.execute(sql_select_query, (f_no,))
        record = mycursor.fetchone()
        print(record)
        sql_Delete_query = """Delete from flight_master where flt_no
= %s"""
        mycursor.execute(sql_Delete_query, (f_no,))
        mydb.commit()

        mycursor.execute(sql_select_query, (f_no,))
        records = mycursor.fetchall()
```

```
      if len(records) == 0:
            print("\nRecord Deleted, Flight number",f_no,"no longer
exists ")
            print()

   except mysql.connector.Error as error:
      print("Failed to delete record from table: {}".format(error))
      print()

   finally:
      print()
      mycursor.close()
##    if (mydb.is_connected()):
##    mydb.close()
##    print("MySQL connection is closed")

def update_flight():
   print("UPDATE FLIGHT".center(100))
   print("----------------------".center(100))
   print()
   f_no=input("Enter Flight Number of Flight to be Updated ")
   print()
   try:
      mycursor = mydb.cursor()
      print("Displaying Flight Details Before Updation: ")
      sql_select_query = """select * from flight_master where flt_no
= %s"""
      mycursor.execute(sql_select_query, (f_no,))
      record = mycursor.fetchone()
      print(record)
      print()
      ch=int(input("Enter 1 to update timings, 2 to update ticket
prices "))
      print()
      if ch==1:
         dep=input("Enter New Departure Time (for eg 1200 hours)
")
         arr=input("Enter New Arrival Time (for eg 1200 hours) ")
```

```
        sql_Update_query = """Update flight_master set departure =
%s, arrival = %s where flt_no = %s"""
        inputdata=(dep,arr,f_no)
        mycursor.execute(sql_Update_query, inputdata)
        mydb.commit()
        mycursor.execute(sql_select_query, (f_no,))
        record = mycursor.fetchone()
        print()
        print(record)
        print("\nRecord successfully updated ")
      elif ch==2:
        cost_eco=float(input("Enter  New Economy Seat Price "))
        cost_prem=float(input("Enter New Premium Seat Price "))
        sql_Update_query = """Update flight_master set cost_eco =
%s, cost_prem = %s where flt_no = %s"""
        inputdata=(cost_eco,cost_prem,f_no)
        mycursor.execute(sql_Update_query, inputdata)
        mydb.commit()
        mycursor.execute(sql_select_query, (f_no,))
        record = mycursor.fetchone()
        print(record)
        print("\nRecord Updated... ")
      else:
        print("Wrong Choice... ")

    except mysql.connector.Error as error:
      print("Failed to Update record: {}".format(error))

    finally:
      print()
      mycursor.close()
##      if (mydb.is_connected()):
##      mydb.close()
##      print("MySQL connection is closed")

def chk_flt():
    print("CHECK ALL FLIGHTS".center(100))
    print("-----------------".center(100))
```

```
    try:
        sql_select_Query = "SELECT * FROM flight_master"
        mycursor = mydb.cursor()
        mycursor.execute(sql_select_Query)
        records = mycursor.fetchall()
        print("Total number of flights in table is: ",
mycursor.rowcount)
        print()
        print("FLIGHT DETAILS".center(100))
        print("--------------".center(100),"\n")
        print("Flt_No".center(6, ' '),"Source_City".center(16, ' '),
"Destination_City".center(16, ' '), "Departure".center(12, '
'),"Arrival".center(12, ' '),"Tot_Eco".center(9, '
'),"Tot_Prem".center(9, ' '), "Eco_Price".center(12, ' '),
"Prem_Price".center(12, ' '))
        print("--------------".ljust(111,'-'),"\n")
        for row in records:
            print(row[0].center(6, ' '),row[1].center(16, ' '),
row[2].center(16, ' '), row[3].center(12, ' '),row[4].center(12, '
'),str(row[5]).center(9, ' '),str(row[6]).center(9, ' '),
str(row[7]).center(12, ' '), str(row[8]).center(12, ' '))
            # print("Price  = ", row[2])
            # print("Purchase date  = ", row[3], "\n")
        print("--------------".ljust(111,'-'),"\n")
        print()
    except Error as e:
        print("Error reading data from MySQL table", e)
        print()
    finally:
        if mydb.is_connected():
            mycursor.close()
            ##mydb.close()
            ##print("MySQL connection is closed")
            print()
            print()

def chk_all_book():
    print("CHECK ALL BOOKINGS".center(100))
```

```
    print("------------------".center(100))
    try:
        sql_select_Query = "SELECT * FROM booking"
        mycursor = mydb.cursor()
        mycursor.execute(sql_select_Query)
        records = mycursor.fetchall()
        print("Total number of bookings in table is: ",
mycursor.rowcount)
        print()
        print("BOOKING DETAILS".center(100))
        print("---------------".center(100),"\n")
        print("BookID".center(8, ' '),"CustomerID".center(12, ' '),
"BookDate".center(10, ' '), "FltNo".center(6, '
'),"SourceCity".center(12, ' '),"Destination".center(12, '
'),"FltDate".center(10, ' '), "Class".center(10, ' '), "Tickets".center(7, '
'), "Price".center(10, ' '), "Departure".center(10, ' '),
"Arrival".center(10, ' '))
        print("---------------".ljust(130,'-'),"\n")
        for row in records:
            print(str(row[0]).center(8, ' '),row[1].center(12, ' '),
str(row[2]).center(10, ' '), row[3].center(6, ' '),row[4].center(12, '
'),str(row[5]).center(12, ' '),str(row[6]).center(10, ' '),
(row[7]).center(10, ' '), str(row[8]).center(7, ' '),
str(row[9]).center(10, ' '), row[10].center(10, ' '), row[11].center(10,
' '))
            # print("Price  = ", row[2])
            # print("Purchase date  = ", row[3], "\n")
        print("---------------".ljust(130,'-'),"\n")
    except Error as e:
        print("Error reading data from MySQL table", e)
        print()
    finally:
        if mydb.is_connected():
            mycursor.close()
            ##mydb.close()
            ##print("MySQL connection is closed")
            print()
            print()
```

```
def chk_date_book():
   print("CHECK BOOKINGS DATEWISE".center(100))
   print("----------------------".center(100))
   enq_date=input("Enter Date(yyyy-mm-dd) ")
   try:
      sql_select_Query = "SELECT * FROM booking where
book_date=%s"
      mycursor = mydb.cursor()
      mycursor.execute(sql_select_Query,(enq_date,))
      records = mycursor.fetchall()
      print("Total number of bookings on ", enq_date, " in table is:
",mycursor.rowcount)
      print()
      print("BOOKING DETAILS".center(100))
      print("--------------".center(100),"\n")
      print("BookID".center(8, ' '),"CustomerID".center(12, ' '),
"BookDate".center(10, ' '), "FltNo".center(6, '
'),"SourceCity".center(12, ' '),"Destination".center(12, '
'),"FltDate".center(10, ' '), "Class".center(10, ' '), "Tickets".center(7, '
'), "Price".center(10, ' '), "Departure".center(10, ' '),
"Arrival".center(10, ' '))
      print("--------------".ljust(130,'-'),"\n")
      for row in records:
         print(str(row[0]).center(8, ' '),row[1].center(12, ' '),
str(row[2]).center(10, ' '), row[3].center(6, ' '),row[4].center(12, '
'),str(row[5]).center(12, ' '),str(row[6]).center(10, ' '),
(row[7]).center(10, ' '), str(row[8]).center(7, ' '),
str(row[9]).center(10, ' '), row[10].center(10, ' '), row[11].center(10,
' '))
         # print("Price  = ", row[2])
         # print("Purchase date  = ", row[3], "\n")
      print("--------------".ljust(130,'-'),"\n")
   except Error as e:
      print("Error reading data from MySQL table", e)
      print()
   finally:
```

```
        if mydb.is_connected():
            mycursor.close()
            ##mydb.close()
            ##print("MySQL connection is closed")
            print()
            print()

def chk_flt_book():
    print("CHECK BOOKINGS FLIGHTWISE".center(100))
    print("------------------------".center(100))
    enq_fltno=input("Enter flight no " )
    try:
        sql_select_Query = "SELECT * FROM booking where
flt_chosen=%s"
        mycursor = mydb.cursor()
        mycursor.execute(sql_select_Query,(enq_fltno,))
        records = mycursor.fetchall()
        print("Total number of bookings in Flight no ", enq_fltno, " is:
",mycursor.rowcount)
        print()
        print("BOOKING DETAILS".center(100))
        print("---------------".center(100),"\n")
        print("BookID".center(8, ' '),"CustomerID".center(12, ' '),
"BookDate".center(10, ' '), "FltNo".center(6, '
'),"SourceCity".center(12, ' '),"Destination".center(12, '
'),"FltDate".center(10, ' '), "Class".center(10, ' '), "Tickets".center(7, '
'), "Price".center(10, ' '), "Departure".center(10, ' '),
"Arrival".center(10, ' '))
        print("--------------".ljust(130,'-'),"\n")
        for row in records:
            print(str(row[0]).center(8, ' '),row[1].center(12, ' '),
str(row[2]).center(10, ' '), row[3].center(6, ' '),row[4].center(12, '
'),str(row[5]).center(12, ' '),str(row[6]).center(10, ' '),
(row[7]).center(10, ' '), str(row[8]).center(7, ' '),
str(row[9]).center(10, ' '), row[10].center(10, ' '), row[11].center(10,
' '))
            # print("Price  = ", row[2])
            # print("Purchase date  = ", row[3], "\n")
```

```
        print("--------------".ljust(130,'-'),"\n")
    except Error as e:
        print("Error reading data from MySQL table", e)
        print()
    finally:
        if mydb.is_connected():
            mycursor.close()
            ##mydb.close()
            ##print("MySQL connection is closed")
            print()
            print()


def chk_flt_avail():
    print("CHECK FLIGHT AVAILABILITY DATEWISE".center(100))
    print("---------------------------------".center(100))
    valid=0

    while valid != 1:
        source=input("Enter Source ")
        source=source.capitalize()
        try:
            mycursor = mydb.cursor()
            sql_select_query1 = """select * from flight_master where
source_city = %s"""
            mycursor.execute(sql_select_query1, (source,))
            records1 = mycursor.fetchall()
            if len(records1) == 0:
                print("\nInvalid Source....Enter Again!!! ")
            else:
                dest=input("Enter Destination ")
                dest=dest.capitalize()
                mycursor = mydb.cursor()
                sql_select_query2 = """select * from flight_master where
destination_city = %s"""
                mycursor.execute(sql_select_query2, (dest,))
                records2 = mycursor.fetchall()
                if len(records2) == 0:
```

```
            print("\nInvalid Destination....Enter Again!!! ")
        else:
            mycursor = mydb.cursor()
            sql_select_query3 = """select * from flight_master
where source_city = %s and destination_city = %s"""
            mycursor.execute(sql_select_query3, (source,dest,))
            records3 = mycursor.fetchall()
            if len(records3) == 0:
                print("No filghts exist between this source destination
pair...Enter again")
            else:
                valid=1
                dt=input("Enter Date of Travel (yyyy-mm-dd)...")
                print()
                sql_select_Query ="SELECT flight_availability.flt_no,
eco_available, prem_available, departure, arrival from
flight_availability, flight_master where
flight_availability.flt_no=flight_master.flt_no and flt_date=%s and
source_city=%s and destination_city=%s"
                mycursor = mydb.cursor()
                mycursor.execute(sql_select_Query,(dt, source,
dest,))
                records = mycursor.fetchall()
                print("Total number of Available Flights are
",mycursor.rowcount)
                print()
                print("          AVAILABLE FLIGHTS ON {} BETWEEN
{} AND {}".format(dt,source.upper(),dest.upper(),))
                print("          ----------------------------------------------
---------,\n")
                print("Flight No".center(11, ' '),"Eco
Available".center(15, ' '), "Prem Available".center(16, ' '),
"Departure".center(12, ' '),"Arrival".center(12, ' '))
                print("--------------".ljust(80,'-'),"\n")
                for row in records:
                    print(str(row[0]).center(11, '
'),str(row[1]).center(15, ' '), str(row[2]).center(16, ' '),
str(row[3]).center(12, ' '),row[4].center(12, ' '))
```

```
                print("--------------".ljust(80,'-'),"\n")
                print()

        except Error as e:
            print("Error reading data from table", e)
            print()
        finally:
            if mydb.is_connected():
                mycursor.close()
                print()
                print()


def chk_all_cust():
    print("CHECK ALL CUSTOMERS".center(100))
    print("------------------".center(100))
    try:
        sql_select_Query = "SELECT * FROM customer"
        mycursor = mydb.cursor()
        mycursor.execute(sql_select_Query)
        records = mycursor.fetchall()
        print("Total number of customers in table is: ",
mycursor.rowcount)
        print()
        print("CUSTOMER DETAILS".center(100))
        print("---------------".center(100),"\n")
        print("CustID".center(8, ' '),"Name".center(15, ' '), "Email
Id".center(20, ' '), "Phone No".center(15, ' '),"Address".center(20, '
'),"Password".center(15, ' '))
        print("--------------".ljust(100,'-'),"\n")
        for row in records:
            print(str(row[0]).center(8, ' '),row[1].center(15, ' '),
row[2].center(20, ' '), str(row[3]).center(15, ' '),row[4].center(20, '
'),row[5].center(15, ' '))
        print("--------------".ljust(100,'-'),"\n")
    except Error as e:
        print("Error reading data from MySQL table", e)
    finally:
```

```
        if mydb.is_connected():
            mycursor.close()
            ##mydb.close()
            ##print("MySQL connection is closed")
            print()
            print()


def del_cust():
    print("DELETE CUSTOMER".center(80))
    print("---------------".center(80))
    enq_custid=input("Enter Customer Username ")
    print()
    try:
        mycursor = mydb.cursor()
        print("Displaying Customer Details Before Deletion: ")
        sql_select_query = """select * from customer where cust_id =
%s"""
        mycursor.execute(sql_select_query, (enq_custid,))
        record = mycursor.fetchone()
        print(record)
        sql_Delete_query = """Delete from customer where cust_id =
%s"""
        mycursor.execute(sql_Delete_query, (enq_custid,))
        mydb.commit()

        mycursor.execute(sql_select_query, (enq_custid,))
        records = mycursor.fetchall()
        if len(records) == 0:
              print("\nRecord Deleted/no longer exists ")

    except mysql.connector.Error as error:
        print("Failed to delete record from table: {}".format(error))
    except:
        print("Failed to delete!!!!Try Again....")

    finally:
        print()
```

```
    mycursor.close()


def admin_enq():
  adm_enq=0
  while adm_enq!=8:
    print("ADMINISTRATOR HELPDESK".center(100))
    print("----------------------".center(100))
    print("1. Check Flight Details")
    print("2. Check All Bookings")
    print("3. Check Bookings made on a particular Date")
    print("4. Check Bookings on a particular Flight")
    print("5. Check Flights Availability on a particular Date")
    print("6. Check Customer Details")
    print("7. Delete Particular Customer")
    print("8. Return to Administrator Menu")
    print()

    adm_enq=int(input("Enter index of option chosen "))
    print()
    if adm_enq==1:
      print()
      chk_flt()
      print()
    elif adm_enq==2:
      print()
      chk_all_book()
    elif adm_enq==3:
      print()
      chk_date_book()
    elif adm_enq==4:
      print()
      chk_flt_book()
      print()
    elif adm_enq==5:
      print()
      chk_flt_avail()
    elif adm_enq==6:
```

```
            print()
            chk_all_cust()
        elif adm_enq==7:
            print()
            del_cust()
        elif adm_enq==8:
            print()
            print("Returning to Administrator Menu")
            print()
            print()
        else:
            print("Invalid choice")
            print()



##def chkphone(ph):
##
##    from twilio.rest import Client
##
##    account_sid = 'ACc10505818c6deb8dcd2fb519f0d5f449'
##    auth_token = '3a596940411377903a4c16c92abaed53'
##    client = Client(account_sid, auth_token)
##    n=random.randint(1111,9999)
##    message = client.messages.create(
##                  body=('OTP for Customer Phone Verification at RB
Airlines Reservation System is %s',n),
##                  from_='+12512552783',
##                  to=ph
##              )
####  print(message.sid)
##    return(n)


def admin():
        password=getpass.getpass(prompt="Password: ")
        if (password=="password" or password=="Password"):
            choice=0
```

```
        while choice!=5:
            print("ADMINISTRTOR MENU".center(100))
            print("-----------------".center(100))
            print("1. Add Flight")
            print("2. Delete Flight")
            print("3. Update Flight")
            print("4. Helpdesk")
            print("5. Return to Main Menu")
            print()

            choice=int(input("Enter index of chosen option "))
            print()

            if choice==1:
                print()
                add_flight()
            elif choice==2:
                print()
                del_flight()
            elif choice==3:
                print()
                update_flight()
            elif choice==4:
                print()
                admin_enq()
            elif choice==5:
                print()
                print("Returning to Main Menu")
                print()
                print()
            else:
                print("Invalid choice")
                print()
        else:
            print("Password Incorrect! ")

def disp_ticket_details(b_id):
    print(" Ticket Information for Booking Id ",b_id)
```

```
    print("\n")
    try:
        mycursor = mydb.cursor()
        sql_select_query1 = """select * from booking where
book_id = %s"""
        mycursor.execute(sql_select_query1, (b_id,))
        record1=mycursor.fetchone()
        sql_select_query2 = """select * from ticketing where
book_id = %s"""
        mycursor.execute(sql_select_query2, (b_id,))
        records2=mycursor.fetchall()
        for row in records2:
            print()
            print("Booking ID:    ",row[1],"              Ticket ID:
",row[0])
            print("----------------------------------------------------------
------------------")
            print("Name:         ",row[2])
            print("Age:         ",row[3])
            print("Gender:       ",row[4])
            print("----------------------------------------------------------
------------------")
            print("Flight Number: ",record1[3],"            Class:
",record1[7])
            print("Source:       ",record1[4],"            Destination:
",record1[5])
            print("Departure:    ",record1[10],"        Arrival:
",record1[11])
            print("Total Price:   ",record1[9],"          Date of Flight:
",record1[6])
            print("----------------------------------------------------------
------------------")
            print()
            print()
            print()
        print("Print your ticket from Customer Enquiry")
        print("Note your Bookind ID and Ticket ID")
        print("Baggage allowed on Economy ticket is 10 Kg")
```

```
        print("Baggage allowed on Premium ticket is 20 Kg")
        print()

    except mysql.connector.Error as error:
        print("Failed to update table {}".format(error))
        print()

    finally:
        print()
        mycursor.close()


def make_booking(flt_chose,dt,travel_class,t_count,amount,usr):
    from datetime import date
    today = date.today()
    try:
        mycursor = mydb.cursor()
        sql_select_query1 = """select * from flight_master where
flt_no = %s"""
        mycursor.execute(sql_select_query1, (flt_chose,))
        record1=mycursor.fetchone()
        sql_select_query2 = """select * from flight_availability
where flt_no = %s and flt_date = %s"""
        mycursor.execute(sql_select_query2, (flt_chose,dt,))
        record2=mycursor.fetchone()
        print(record2)
        sql_select_query3 = """select * from booking"""
        mycursor.execute(sql_select_query3)
        record3=mycursor.fetchall()
        b_id=(record3[len(record3)-1][0]) + 1
        sql_insert_query1 = """INSERT INTO booking (book_id,
customer_id, book_date, flt_chosen,source_city,destination_city,
flt_date, put_class, tickets_count,price,departure,arrival)
        VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s) """
        recordTuple = (b_id, usr, today, flt_chose, record1[1],
record1[2], dt, travel_class, t_count, amount, record1[3],
record1[4])
        mycursor.execute(sql_insert_query1, recordTuple)
```

```
        print("Your Booking is confirmed:\n\n")
        print("Your Booking Details are: \n")
        print("Booking Id      ", b_id)
        print("Username        ", usr)
        print("Booking Date    ", today)
        print("Flight Number   ", flt_chose)
        print("Source City     ", record1[1])
        print("Destination City ", record1[2])
        print("Date of Travel  ", dt)
        print("Class           ", travel_class)
        print("No of Tickets   ", t_count)
        print("Amount          Rs.", amount)
        print("Departure       ", record1[3])
        print("Arrival         ", record1[4])
        print()
        print()
        print("Enter Following Passenger Details\n")
        print()
        for j in range(1,t_count+1):
            print("Enter name on ticket ",j)
            name=input("")
            print("Enter age of person on ticket ",j)
            age=int(input(""))
            print("Enter gender for ticket(M/F) ",j)
            gender=input("")
            gender=gender.capitalize()
            t_id=j
            sql_insert_query2 = """INSERT INTO ticketing (ticket_id,
book_id, name, age, gender) VALUES (%s,%s,%s,%s,%s)"""
            recordTuple = (t_id, b_id, name, age, gender)
            mycursor.execute(sql_insert_query2, recordTuple)
        new_eco=int(record2[2])-t_count
        new_prem=int(record2[3])-t_count
        if travel_class == 'Economy':
            mycursor = mydb.cursor()
            sql_upd_query1 = """Update flight_availability set
eco_available = %s where flt_no = %s and flt_date = %s"""
```

```
            mycursor.execute(sql_upd_query1,
(new_eco,flt_chose,dt,))
        elif travel_class == 'Premium':
            mycursor = mydb.cursor()
            sql_upd_query1 = """Update flight_availability set
prem_available = %s where flt_no = %s and flt_date = %s"""
            mycursor.execute(sql_upd_query1,
(new_prem,flt_chose,dt,))
        print()
        mydb.commit()

        disp_ticket_details(b_id)

    except mysql.connector.Error as error:
        print("Failed to update table {}".format(error))
        print()

    finally:
        print()
        mycursor.close()


def book_ticket(usr):
    print("TICKET BOOKING PANEL".center(80))
    print("--------------------".center(80))
    number_variable=0
    print()
    valid=0
    while valid != 1:
        source=input("Enter Source ")
        source=source.capitalize()
        try:
            mycursor = mydb.cursor()
            sql_select_query1 = """select * from flight_master where
source_city = %s"""
            mycursor.execute(sql_select_query1, (source,))
            records1 = mycursor.fetchall()
            if len(records1) == 0:
```

```
            print("\nInvalid Source....Enter Again!!! ")
        else:
            dest=input("Enter Destination ")
            dest=dest.capitalize()
            mycursor = mydb.cursor()
            sql_select_query2 = """select * from flight_master where
destination_city = %s"""
            mycursor.execute(sql_select_query2, (dest,))
            records2 = mycursor.fetchall()
            if len(records2) == 0:
                print("\nInvalid Destination....Enter Again!!! ")
            else:
                mycursor = mydb.cursor()
                sql_select_query3 = """select * from flight_master
where source_city = %s and destination_city = %s"""
                mycursor.execute(sql_select_query3, (source,dest,))
                records3 = mycursor.fetchall()
                if len(records3) == 0:
                    print("No filghts exist between this source destination
pair...Enter again")
                else:
                    valid=1
                    dt=input("Enter Date of Travel (yyyy-mm-dd)...")
                    print()
                    tickets_count=int(input("Enter no. of passengers "))
                    print()
                    class_chosen=int(input("Enter preffered class(1 for
Economy / 2 for Premium) "))
                    print()
                    if class_chosen!=1 and class_chosen!=2 :
                        print("Invalid Class...enter again")
                        class_chosen=int(input("Enter preffered class(1 for
Economy / 2 for Premium) "))
                    y=0
                    while y!=1:
                        for row in records3:
                            sql_select_query4 = """select * from
flight_availability where flt_no = %s and flt_date = %s"""
```

```
                    mycursor.execute(sql_select_query4, (row[0],
dt,))
                    records4 = mycursor.fetchall()
                   # records5 = mycursor.fetchone()
                    a=len(records4)-1
                    if len(records4) == 0:
                       continue
                    else:
                       if class_chosen==1:
                          if tickets_count <= int(records4[a][2]):
                             number_variable +=1
                             print()
                             print("Details of available
flight",number_variable,"are as follows:")
                             print("\nFlight Number is     ", row[0])
                             print("Source City is       ", row[1])
                             print("Destination City is  ", row[2])
                             print("Departure is         ", row[3])
                             print("Arrival is           ", row[4])
                             print("Cost per ticket is   ", row[7])
                             tot_cost=tickets_count*float(row[7])
                             print("Seats available      ",records4[a][2])
                             y=1
                       elif class_chosen==2:
                          if tickets_count <= int(records4[a][3]):
                             number_variable +=1
                             print()
                             print("Details of available
flight",number_variable,"are as follows:")
                             print("\nFlight Number is     ", row[0])
                             print("Source City is       ", row[1])
                             print("Destination City is  ", row[2])
                             print("Departure is         ", row[3])
                             print("Arrival is           ", row[4])
                             print("Cost per ticket is   ", row[8])
                             tot_cost=tickets_count*float(row[8])
                             print("Seats available      ",records4[a][3])
                             y=1
```

```
                        else:
                            print()
                    if y==0:
                        print("No flights available...change date")
                        ch=input("Do you want to try another date (y/n)?
")
                        if ch=='y':
                            dt=input("Enter Date of Travel (yyyy-dd-
mm)...")
                        else:
                            y=1
                    else:
                        print()
                        print("Do you want to book any of the available
flights?")
                        print("1. Yes")
                        print("2. No")
                        print()
                        want =input("Enter index of answer ")
                        print()
                        if int(want) == 1:
                            put_class=""
                            if class_chosen==1:
                                put_class="Economy"
                            else:
                                put_class="Premium"
                            flt_chosen=input("Enter flight number of the
flight chosen ")
                            flt_chosen=flt_chosen.capitalize()

make_booking(flt_chosen,dt,put_class,tickets_count,tot_cost,usr)
                        else:
                            print("Try Again!!!!!!")
                            print("Returning to Customer Menu....")

        except mysql.connector.Error as error:
            print("Failed to select records from table: {}".format(error))
```

```
        finally:
            print()
            mycursor.close()


def cancel_ticket():
    print("CANCEL TICKET".center(100))
    print("-------------".center(100))
    print()
    b_id=input("Enter Booking Id of Ticket...")
    print()
    try:
        mycursor = mydb.cursor()
        sql_select_query = """select * from booking where book_id =
%s"""
        mycursor.execute(sql_select_query, (b_id,))
        record = mycursor.fetchone()
        print("Your Booking Details are: \n")
        print("Booking Id      ", record[0])
        print("Username        ", record[1])
        print("Booking Date    ", record[2])
        print("Flight Number   ", record[3])
        print("Source City     ", record[4])
        print("Destination City ", record[5])
        print("Date of Travel  ", record[6])
        print("Class           ", record[7])
        print("No of Tickets   ", record[8])
        print("Amount          Rs.", record[9])
        print("Departure       ", record[10])
        print("Arrival         ", record[11])
        print()
        print()
        travel_class=record[7]
        flt_book=record[3]
        dt=record[6]
        amount=float(record[9])
        t_count=int(record[8])
        print("1.Cancel Entire Booking")
```

```
        print("2.Cancel Particular Ticket on this Booking\n")
        ch=int(input("Enter choice..."))
        if ch==1:
            sql_Delete_query1 = """Delete from ticketing where
book_id = %s"""
            mycursor.execute(sql_Delete_query1, (b_id,))
            sql_Delete_query2 = """Delete from booking where book_id
= %s"""
            mycursor.execute(sql_Delete_query2, (b_id,))
            if travel_class=="Economy":
                sql_Update_query1 = """Update flight_availability set
eco_available = eco_available + %s where flt_no = %s and flt_date
= %s"""
                mycursor.execute(sql_Update_query1,
(t_count,flt_book,dt,))
            else:
                sql_Update_query2 = """Update flight_availability set
prem_available = prem_available + %s where flt_no = %s and
flt_date = %s"""
                mycursor.execute(sql_Update_query2,
(t_count,flt_book,dt,))
            print()
            print("Your Booking with Booking Id ", b_id, "is cancelled")
            print()
        elif ch==2:
            cost_per_ticket=amount/t_count
            rev_amount=cost_per_ticket * (t_count-1)
            t_id=int(input("Enter Ticket Id to be Cancelled..."))
            sql_Delete_query3 = """Delete from ticketing where
book_id = %s and ticket_id = %s"""
            mycursor.execute(sql_Delete_query3, (b_id,t_id,))
            sql_Update_query3 = """Update booking set tickets_count =
%s where book_id = %s"""
            mycursor.execute(sql_Update_query3, (t_count-1,b_id,))
            sql_Update_query4 = """Update booking set price = %s
where book_id = %s"""
            mycursor.execute(sql_Update_query4, (rev_amount,b_id,))
            if travel_class=="Economy":
```

```
            sql_Update_query5 = """Update flight_availability set
eco_available = eco_available + 1 where flt_no = %s and flt_date =
%s"""
            mycursor.execute(sql_Update_query5, (flt_book,dt,))
         else:
            sql_Update_query6 = """Update flight_availability set
prem_available = prem_available + 1 where flt_no = %s and
flt_date = %s"""
            mycursor.execute(sql_Update_query6, (flt_book,dt,))
         print("Your Ticket with Booking Id ", b_id, "and Ticket Id ",
t_id, "is cancelled...")
         print()
      else:
         print("Wrong Choice....Try Again!!!")

      mydb.commit()

   except mysql.connector.Error as error:
      print("Failed to Update table: {}".format(error))
      print()
   except:
      print("Failed, Try Again!!!")
   finally:
      print()
      mycursor.close()

def chk_cust_book():
   print("CHECK BOOKINGS".center(100))
   print("--------------".center(100))
   print()
   b_id=input("Enter Booking Id...")
   print()
   try:
      mycursor = mydb.cursor()
      sql_select_query = """select * from booking where book_id =
%s"""
      mycursor.execute(sql_select_query, (b_id,))
      record = mycursor.fetchone()
```

```
        print("Your Booking Details are: \n")
        print("Booking Id       ", record[0])
        print("Username         ", record[1])
        print("Booking Date     ", record[2])
        print("Flight Number    ", record[3])
        print("Source City      ", record[4])
        print("Destination City ", record[5])
        print("Date of Travel   ", record[6])
        print("Class            ", record[7])
        print("No of Tickets    ", record[8])
        print("Amount           Rs.", record[9])
        print("Departure        ", record[10])
        print("Arrival          ", record[11])
        print()
        print()

    except:
        print("Failed to fetch booking data!!!!")

    finally:
        print()
        mycursor.close()

def print_tkt():
    print("PRINT YOUR TICKET".center(100))
    print("------------------".center(100))


    enquire_bookid=int(input("Enter your Booking ID "))
    enquire_ticketid=int(input("Enter your Ticket ID "))
    wrong_tckt_var=0

    try:
        f=open("ticket.txt","w")
        mycursor = mydb.cursor()
        sql_select_query = """select
name,age,gender,flt_chosen,source_city,destination_city,flt_date,
put_class,price,departure,arrival from booking,ticketing where
```

```
booking.book_id = %s and ticket_id= %s and
booking.book_id=ticketing.book_id"""
      mycursor.execute(sql_select_query,
(enquire_bookid,enquire_ticketid,))
      record = mycursor.fetchone()

      print("Your Ticket Details are as follows:\n\n\n")


print("**************************************************
**********************************************")
      print("                                    TICKET")

print("**************************************************
**********************************************\n")
      print("Booking ID:    ",enquire_bookid,"
Ticket ID:     ",enquire_ticketid)
      print("----------------------------------------------------------------
----------------------------")
      print("Name:         ",record[0])
      print("Age:          ",record[1])
      print("Gender:       ",record[2])
      print("----------------------------------------------------------------
----------------------------")
      print("Flight Number:        ",record[3].rjust(12),"
Class:        ",record[7])
      print("Source:               ",record[4].rjust(12),"
Destination:        ",record[5])
      print("Departure:            ",record[9].rjust(12),"
Arrival:        ",record[10])
      print("Date of Flight:       ",record[6],"               Total Price:
",record[8])
      print("----------------------------------------------------------------
----------------------------\n")
      print("Reporting time is 2 Hours before departure")
      if (record[7]=="Economy"):
         print("Baggage allowed on your ticket is 10 Kg")
         print("Cabin Meal not included in your ticket")
```

```
        else:
            print("Baggage allowed on your ticket is 20 Kg")
            print("Cabin Meal included in your ticket")

print("**************************************************
**************************************************\n\n\n")
        print("The above ticket has been saved on your system for
printing purpose...\n\n")



f.write("**************************************************
**************************************************\n")
        f.write("                                    TICKET\n")

f.write("**************************************************
**************************************************\n")
        f.write("\n")
        ab=str("Booking ID:    " + str(enquire_bookid) + "
Ticket ID:     " + str(enquire_ticketid) + "\n")
        f.write(ab)
        ab=str("-------------------------------------------------------------
-----------------------------\n")
        f.write(ab)
        ab=str("Name:         " + record[0] +"\n")
        f.write(ab)
        ab=str("Age:          " + str(record[1]) +"\n")
        f.write(ab)
        ab=str("Gender:       " + record[2] + "\n")
        f.write(ab)
        ab=str("-------------------------------------------------------------
-----------------------------\n")
        f.write(ab)
        ab=str("Flight Number:         " + record[3] + "
Class:        " + record[7] + "\n")
        f.write(ab)
        ab=str("Source:               " + record[4] + "
Destination:        " + record[5] + "\n")
```

```
        f.write(ab)
        ab=str("Departure:            " + record[9] + "                Arrival:
" + record[10] + "\n")
        f.write(ab)
        ab=str("Total Price:          " + str(record[8]) + "             Date
of Flight:         " + str(record[6]) + "\n")
        f.write(ab)
        ab=str("-------------------------------------------------------------------
------------------------------\n")
        f.write(ab)
        f.write("Reporting time is 2 Hours before departure\n")
        if (record[7]=="Economy"):
            f.write("Baggage allowed on your ticket is 10 Kg\n")
            f.write("Cabin Meal not included in your ticket\n")
        else:
            f.write("Baggage allowed on your ticket is 20 Kg\n")
            f.write("Cabin Meal included in your ticket\n")

f.write("**************************************************
**************************************************\n")

    except mysql.connector.Error as error:
        print("Failed to fetch booking data: {}".format(error))
    except:
        print("No Ticket!!!!Try Again.....")

    finally:
        print()
        f.close()
        mycursor.close()

def food_bag():
    print("FOOD/BAGGAGE GUIDE".center(100))
    print("-----------------".center(100))
    print()
    print("Cabin Meal is include in Premium ticket")
    print("Baggage allowed on Premium ticket is 20 Kg\n")
    print("Cabin Meal is not included Economy ticket")
```

```
    print("Baggage allowed on Economy ticket is 10 Kg\n\n")

def cust_enq():

    choice=0
    while choice!=5:
        print("CUSTOMER ENQUIRIES".center(100))
        print("------------------".center(100))
        print("1. Check Flight Availability")
        print("2. Check Your Booking")
        print("3. Print Ticket")
        print("4. Your Food/Baggage Guide")
        print("5. Return to Customer Menu")
        print()
        choice=int(input("Enter index of chosen option "))
        print()
        if choice==1:
            print()
            chk_flt_avail()
        elif choice==2:
            print()
            chk_cust_book()
        elif choice==3:
            print()
            print_tkt()
        elif choice==4:
            print()
            food_bag()
        elif choice==5:
            print()
            print("Returning to Customer Menu")
            print()
            print()
        else:
            print("Invalid choice")
            print()

def customer():
```

```
    #print("Enter username and password")
    print("If new user, please create Customer Account first")
    username=input("\nEnter Username: ")
    try:
        mycursor = mydb.cursor()
        sql_select_query = """select passwd from customer where
cust_id = %s"""
        mycursor.execute(sql_select_query, (username,))
        records = mycursor.fetchall()
        if len(records) == 0:
            print("\nUsername does not exists...Try Again!!! ")
            print()
        else:
            passw=getpass.getpass(prompt="Enter Password: ")
            print()
        for row in records:
            if row[0]==passw:
                choice=0
                while choice!=4:
                    print("CUSTOMER MENU".center(100))
                    print("-------------".center(100))
                    print("1. Book Ticket")
                    print("2. Cancel Ticket")
                    print("3. Enquire")
                    print("4. Return to Main Menu")
                    print()
                    choice=int(input("Enter index of chosen option "))
                    print()
                    if choice==1:
                        print()
                        book_ticket(username)
                    elif choice==2:
                        print()
                        cancel_ticket()
                    elif choice==3:
                        print()
                        cust_enq()
                    elif choice==4:
```

```
                print()
                print("Returning to Main Menu")
                print()
                print()
            else:
                print("Invalid choice")
                print()
        else:
            print("Password Incorrect")
            print()

    except mysql.connector.Error as error:
        print("Failed to verify Customer Account: {}".format(error))

    finally:
        print()
        mycursor.close()


def cust_acct():
    print("CREATE NEW CUSTOMER ACCOUNT".center(80))
    print("--------------------------".center(80))
    print("Customer Details:  ")
    print()
    na=input("Enter Name ")
    email=input("Enter Email Id ")
    phone=int(input("Enter 10 digit Mobile Number "))
##    phone='+91'+ phone
    address=input("Enter Address ")
    username=input("Enter Username for this Account ")
    passw=input("Enter Password for this Account ")
##    x=chkphone(phone)
##    otp=input("Enter OTP recived on your phone for
verification")
##    if x==otp:
##        print("phone number verified")
    try:
```

```
      mySql_insert_query = """INSERT INTO customer (cust_id,
name, emailid, phone_no, address, passwd)
        VALUES (%s,%s,%s,%s,%s,%s) """
      recordTuple = (username, na, email, phone, address, passw )
      mycursor = mydb.cursor()
      mycursor.execute(mySql_insert_query, recordTuple)
      mydb.commit()
      print(mycursor.rowcount, "Record inserted successfully into
Customer table")
      mycursor.close()

   except mysql.connector.Error as error:
      print("Failed to insert record into table {}".format(error))

   finally:
      print()
   ##if (mydb.is_connected()):
         ##connection.close()
   ##print("MySQL connection is closed")
##    else:
##       print("Wrong OTP....Account creation aborted!!")
##       print()




mydb=mysql.connector.connect(host="localhost",user="root",pas
sword="root",database="RB_AIRLINES")

index=0

print("*"*150)
print("*"*150)
print()
print()
print()
print()
print("                                                *       ")
```

```
print("                                        *****    ")
print("                                       *******   ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                     ***********  ")
print("                                    **************  ")
print("              Welcome to                  ********************
")
print("               __    __
*****************************  ")
print("              || \\\ || \\\\
************************************  ")
print("              || // || //
*******************************************  ")
print("              ||==   ||===       ************    *********
************ ")
print("              || \\\  ||  \\\\     ********      *********
******** ")
print("              || \\\ ||___//     ***          *********
*** ")
print("                                 *          *********          * ")
print("              Airline System                    *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                      *********  ")
print("                                     **********  ")
print("                                    ************  ")
print("                                   *********  *********  ")
```

```
print("                                        ********      ******** ")
print("                                       *******        ******* ")
print("                                      *****            ***** ")
print("                                      **                ** ")
print()
print()
print()
print()
print("*"*150)
print("*"*150)
print()
print("LOADING...", end="", flush=True)
for i in range(0,140):
    time.sleep(0.038)
    print("|", end="", flush=True)
clear()

while index!=4:
    print("*"*150)
    print()
    print("WELCOME TO RB AIRLINE SYSTEM".center(120))
    print("-------------------------".center(120))
    print("*"*150)
    print()
    print("1. Administrator")
    print("2. Customer Login ")
    print("3. Create Customer Account")
    print("4. EXIT")
    print()

    index =int(input("Enter Index (1 for Admin, 2 for Customer, 3 for
Creating new customer account) "))
    print()
    if index == 1:
        print()
        admin()

    elif index==2:
```

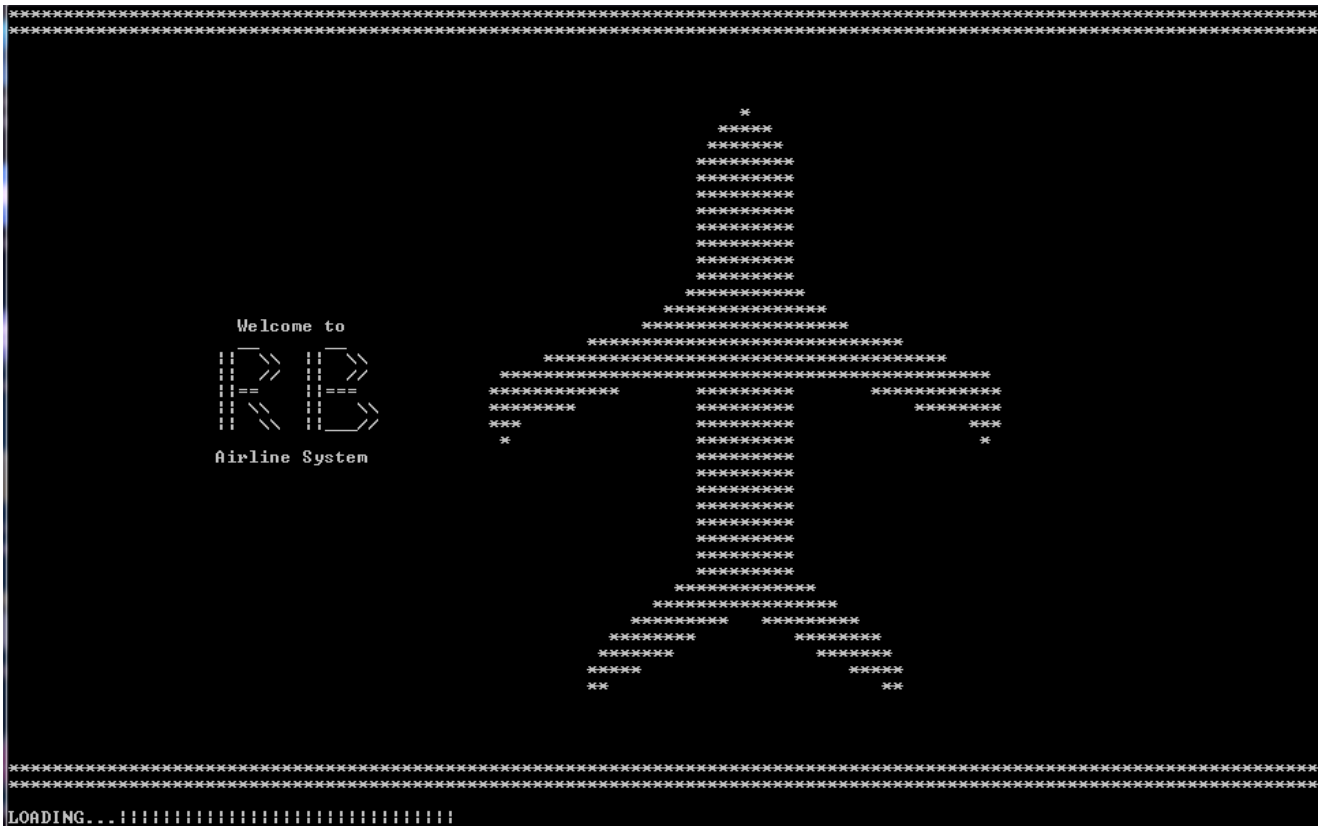
```
    print()
    customer()

elif index==3:
    print()
    cust_acct()

elif index==4:
    print()
    print("HAVE A GOOD DAY".center(110))
    print("---------------".center(110))
    time.sleep(2)
    exit

else:
    print("INVALID CHOICE")
```

OUTPUT SCREENS




```
                                    WELCOME TO RB AIRLINE SYSTEM
                                    ----------------------------

1. Administrator
2. Customer Login
3. Create Customer Account
4. EXIT

Enter Index (1 for Admin, 2 for Customer, 3 for Creating new customer account) 1

Password:
                                        ADMINISTRTOR MENU
                                        -----------------

1. Add Flight
2. Delete Flight
3. Update Flight
4. Helpdesk
5. Return to Main Menu

Enter index of chosen option 1
```

```
                                        ADD FLIGHT
                                       ------------
Enter New Flight Details:

Enter Flight Number F030
Enter Source City Bangalore
Enter Destination City Hyderabad
Enter Departure Time (for eg 1200 hours )1300 hours
Enter Arrival Time (for eg 1200 hours) 1500 hours
Enter Total Number of Economy Seats in Flight 25
Enter Total Number of Premium Seats in Flight 25
Enter  Economy Seat Price 800
Enter  Premium Seat Price 1200

1 Record inserted successfully into Flight Master table


                                     ADMINISTRTOR MENU
                                     -----------------
1. Add Flight
2. Delete Flight
3. Update Flight
4. Helpdesk
5. Return to Main Menu

Enter index of chosen option 3


                                      UPDATE FLIGHT
                                     ---------------------

Enter Flight Number of Flight to be Updated F030

Displaying Flight Details Before Updation:
('F030', 'Bangalore', 'Hyderabad', '1300 hours', '1500 hours', 25, 25, 800.0, 1200.0)

Enter 1 to update timings, 2 to update ticket prices 2

Enter  New Economy Seat Price 800
Enter New Premium Seat Price 1500
('F030', 'Bangalore', 'Hyderabad', '1300 hours', '1500 hours', 25, 25, 800.0, 1500.0)

Record Updated...
```

```
                                    ADMINISTRTOR MENU
                                    -----------------
1. Add Flight
2. Delete Flight
3. Update Flight
4. Helpdesk
5. Return to Main Menu

Enter index of chosen option 4

                                  ADMINISTRATOR HELPDESK
                                  ----------------------
1. Check Flight Details
2. Check All Bookings
3. Check Bookings made on a particular Date
4. Check Bookings on a particular Flight
5. Check Flights Availability on a particular Date
6. Check Customer Details
7. Delete Particular Customer
8. Return to Administrator Menu

Enter index of option chosen 1

                                    CHECK ALL FLIGHTS
                                    -----------------
Total number of flights in table is:  14
                                      FLIGHT DETAILS
                                      --------------
Flt_No   Source_City    Destination_City  Departure     Arrival     Tot_Eco   Tot_Prem  Eco_Price    Prem_Price
-------------------------------------------------------------------------------------------------------------

 F001      Chennai          Delhi        1200 hours   1600 hours      30        30       500.0        1000.0
 F002       Delhi           Mumbai       0900 hours   1100 hours      30        30       600.0        1200.0
 F003       Delhi          Chennai       1700 hours   2100 hours      30        30       500.0        1000.0
 F004       Mumbai          Delhi        1200 hours   1400 hours      30        30       600.0        1200.0
 F005       Mumbai         Chennai       2000 hours   0100 hours      30        30       500.0        1000.0
 F006       Delhi           Mumbai       0500 hours   0700 hours      30        30       500.0        1000.0
 F007      Chennai          Mumbai       0500 hours   1000 hours      30        30       500.0        1000.0
 F008       Mumbai          Delhi        0800 hours   1100 hours      30        30       500.0        1000.0
 F009       Delhi          Chennai       1600 hours   2000 hours      30        30       400.0        9000.0
 F010       Delhi           Mumbai       1200 hours   1400 hours      30        30       500.0        1000.0
 F011      Chennai          Delhi        2100 hours   0200 hours      30        30       400.0        900.0
 F012       Mumbai          Delhi        1000 hours   1200 hours      30        30      10000.0      20000.0
 F013       Delhi            Agra        1230 hours   1300 hours      45        45       500.0        1500.0
 F030     Bangalore        Hyderabad      1300 hours   1500 hours      25        25       800.0        1500.0
-------------------------------------------------------------------------------------------------------------
```

```
                                    CHECK ALL BOOKINGS
                                    ------------------
Total number of bookings in table is:  1
                                      BOOKING DETAILS
                                      ---------------
 BookID   CustomerID   BookDate  FltNo   SourceCity  Destination   FltDate     Class     Tickets   Price    Departure   A
-------------------------------------------------------------------------------------------------------------------------

   1         rad      2020-10-25  F010     Delhi        Mumbai    2020-07-29  Economy      1      500.0    1200 hours 14
-------------------------------------------------------------------------------------------------------------------------



                                  ADMINISTRATOR HELPDESK
                                  ----------------------
1. Check Flight Details
2. Check All Bookings
3. Check Bookings made on a particular Date
4. Check Bookings on a particular Flight
5. Check Flights Availability on a particular Date
6. Check Customer Details
7. Delete Particular Customer
8. Return to Administrator Menu

Enter index of option chosen 5

                               CHECK FLIGHT AVAILABILITY DATEWISE
                               ----------------------------------
Enter Source Delhi
Enter Destination Mumbai
Enter Date of Travel (yyyy-mm-dd)...2020-07-28

Total number of Available Flights are  3

                AVAILABLE FLIGHTS ON 2020-07-28 BETWEEN DELHI AND MUMBAI
                --------------------------------------------------------

 Flight No   Eco Available   Prem Available   Departure     Arrival
-------------------------------------------------------------------------------

    F002           25              25         0900 hours   1100 hours
    F006           25              25         0500 hours   0700 hours
    F010           25              25         1200 hours   1400 hours
-------------------------------------------------------------------------------
```

```
                                ADMINISTRATOR HELPDESK
                                ----------------------
1. Check Flight Details
2. Check All Bookings
3. Check Bookings made on a particular Date
4. Check Bookings on a particular Flight
5. Check Flights Availability on a particular Date
6. Check Customer Details
7. Delete Particular Customer
8. Return to Administrator Menu

Enter index of option chosen 6

                                  CHECK ALL CUSTOMERS
                                  -------------------
Total number of customers in table is:  6
                                    CUSTOMER DETAILS
                                    ----------------

 CustID        Name             Email Id          Phone No          Address            Password
------------------------------------------------------------------------------------------------

  rad         Radhika             rad              27231             hrjef              rad
Radhika       Radhika          rads@hjiws        9810838276        hgwjkkdjwch         radhika
  Rads        Radhika             rads           098019272           2hdcbiva           Rads
  rak1         rakesh          rcb@bird.in       9810001030       6/1 jaidev park       rak1
 rb123     Roli Bansal          roli@123          9876543210        Jaidev park         rb456
  roli           R                 er             +919876              fgh              roli
------------------------------------------------------------------------------------------------
```

```
                                DELETE CUSTOMER
                                ---------------
Enter Customer Username rb123

Displaying Customer Details Before Deletion:
('rb123', 'Roli Bansal', 'roli@123', '9876543210', 'Jaidev park', 'rb456')

Record Deleted/no longer exists
```

```
1. Administrator
2. Customer Login
3. Create Customer Account
4. EXIT

Enter Index (1 for Admin, 2 for Customer, 3 for Creating new customer account) 3


                        CREATE NEW CUSTOMER ACCOUNT
                        ---------------------------
Customer Details:

Enter Name Ruhani
Enter Email Id ruh@ABC.com
Enter 10 digit Mobile Number 1234567891
Enter Address New Delhi-XX
Enter Username for this Account Ruhani
Enter Password for this Account mypass
1 Record inserted successfully into Customer table

*****************************************************************************************************************************

                                          WELCOME TO RB AIRLINE SYSTEM
                                          ----------------------------
*****************************************************************************************************************************

1. Administrator
2. Customer Login
3. Create Customer Account
4. EXIT

Enter Index (1 for Admin, 2 for Customer, 3 for Creating new customer account) 2


If new user, please create Customer Account first

Enter Username: Ruhani
Enter Password:

                                       CUSTOMER MENU
                                       -------------

1. Book Ticket
2. Cancel Ticket
3. Enquire
4. Return to Main Menu
```

```
                          TICKET BOOKING PANEL
                          --------------------

Enter Source Delhi
Enter Destination Mumbai
Enter Date of Travel (yyyy-mm-dd)...2020-07-28

Enter no. of passengers 2

Enter preffered class(1 for Economy / 2 for Premium) 1


Details of available flight 1 are as follows:

Flight Number is      F002
Source City is        Delhi
Destination City is   Mumbai
Departure is          0900 hours
Arrival is            1100 hours
Cost per ticket is    600.0
Seats available       25

Details of available flight 2 are as follows:

Flight Number is      F006
Source City is        Delhi
Destination City is   Mumbai
Departure is          0500 hours
Arrival is            0700 hours
Cost per ticket is    500.0
Seats available       25

Details of available flight 3 are as follows:

Flight Number is      F010
Source City is        Delhi
Destination City is   Mumbai
Departure is          1200 hours
Arrival is            1400 hours
Cost per ticket is    500.0
Seats available       25

Do you want to book any of the available flights?
1. Yes
2. No

Enter index of answer 1
```

```
Enter flight number of the flight chosen F006
('F006', datetime.date(2020, 7, 28), 25, 25)
Your Booking is confirmed:


Your Booking Details are:

Booking Id          2
Username            Ruhani
Booking Date        2020-11-27
Flight Number       F006
Source City         Delhi
Destination City    Mumbai
Date of Travel      2020-07-28
Class               Economy
No of Tickets       2
Amount              Rs. 1000.0
Departure           0500 hours
Arrival             0700 hours


Enter Following Passenger Details


Enter name on ticket  1
Ruhani
Enter age of person on ticket  1
17
Enter gender for ticket(M/F)  1
F
Enter name on ticket  2
Rohan
Enter age of person on ticket  2
27
Enter gender for ticket(M/F)  2
M

 Ticket Information for Booking Id  2
```

```
Booking ID:     2                       Ticket ID:       1
--------------------------------------------------------------------------------
Name:           Ruhani
Age:            17
Gender:         F
--------------------------------------------------------------------------------
Flight Number:  F006                    Class:           Economy
Source:         Delhi                   Destination:      Mumbai
Departure:      0500 hours                  Arrival:         0700 hours
Total Price:    1000.0                   Date of Flight:  2020-07-28
--------------------------------------------------------------------------------



Booking ID:     2                       Ticket ID:       2
--------------------------------------------------------------------------------
Name:           Rohan
Age:            27
Gender:         M
--------------------------------------------------------------------------------
Flight Number:  F006                    Class:           Economy
Source:         Delhi                   Destination:      Mumbai
Departure:      0500 hours                  Arrival:         0700 hours
Total Price:    1000.0                   Date of Flight:  2020-07-28
--------------------------------------------------------------------------------



Print your ticket from Customer Enquiry
Note your Bookind ID and Ticket ID
Baggage allowed on Economy ticket is 10 Kg
Baggage allowed on Premium ticket is 20 Kg
```

```
                                    CUSTOMER MENU
                                    -------------
1. Book Ticket
2. Cancel Ticket
3. Enquire
4. Return to Main Menu

Enter index of chosen option 3

                                  CUSTOMER ENQUIRIES
                                  ------------------
1. Check Flight Availability
2. Check Your Booking
3. Print Ticket
4. Your Food/Baggage Guide
5. Return to Customer Menu

Enter index of chosen option 2

                                    CHECK BOOKINGS
                                    --------------

Enter Booking Id...1

Your Booking Details are:

Booking Id          1
Username            rad
Booking Date        2020-10-25
Flight Number       F010
Source City         Delhi
Destination City    Mumbai
Date of Travel      2020-07-29
Class               Economy
No of Tickets       1
Amount              Rs. 500.0
Departure           1200 hours
Arrival             1400 hours
```

```
                                        PRINT YOUR TICKET
                                        -----------------
Enter your Booking ID 2
Enter your Ticket ID 1
Your Ticket Details are as follows:


********************************************************************************************************************
                                                TICKET
********************************************************************************************************************

Booking ID:     2                                                               Ticket ID:      1
--------------------------------------------------------------------------------------------------------------------
Name:           Ruhani
Age:            17
Gender:         F
--------------------------------------------------------------------------------------------------------------------
Flight Number:                  F006                            Class:          Economy
Source:                         Delhi                       Destination:        Mumbai
Departure:                  0500 hours                          Arrival:        0700 hours
Date of Flight:             2020-07-28                      Total Price:        1000.0
--------------------------------------------------------------------------------------------------------------------

Reporting time is 2 Hours before departure
Baggage allowed on your ticket is 10 Kg
Cabin Meal not included in your ticket
********************************************************************************************************************


The above ticket has been saved on your system for printing purpose...


1. Check Flight Availability
2. Check Your Booking
3. Print Ticket
4. Your Food/Baggage Guide
5. Return to Customer Menu

Enter index of chosen option 4

                                        FOOD/BAGGAGE GUIDE
                                        ------------------

Cabin Meal is include in Premium ticket
Baggage allowed on Premium ticket is 20 Kg

Cabin Meal is not included Economy ticket
Baggage allowed on Economy ticket is 10 Kg
```

```
********************************************************************************************************************
                                                TICKET
********************************************************************************************************************

Booking ID:     2                                                               Ticket ID:      1
--------------------------------------------------------------------------------------------------------------------
Name:           Ruhani
Age:            17
Gender:         F
--------------------------------------------------------------------------------------------------------------------
Flight Number:          F006                            Class:          Economy
Source:                 Delhi                       Destination:        Mumbai
Departure:              0500 hours                      Arrival:        0700 hours
Total Price:            1000.0                  Date of Flight:         2020-07-28
--------------------------------------------------------------------------------------------------------------------
Reporting time is 2 Hours before departure
Baggage allowed on your ticket is 10 Kg
Cabin Meal not included in your ticket
********************************************************************************************************************
```

```
                                        CUSTOMER MENU
                                        -------------
1. Book Ticket
2. Cancel Ticket
3. Enquire
4. Return to Main Menu

Enter index of chosen option 2

                                        CANCEL TICKET
                                        -------------

Enter Booking Id of Ticket...2

Your Booking Details are:

Booking Id          2
Username            Ruhani
Booking Date        2020-11-27
Flight Number       F006
Source City         Delhi
Destination City    Mumbai
Date of Travel      2020-07-28
Class               Economy
No of Tickets       2
Amount              Rs. 1000.0
Departure           0500 hours
Arrival             0700 hours


1.Cancel Entire Booking
2.Cancel Particular Ticket on this Booking

Enter choice...2
Enter Ticket Id to be Cancelled...2
Your Ticket with Booking Id  2 and Ticket Id  2 is cancelled...
```

```
Password:
                                        ADMINISTRTOR MENU
                                        -----------------
1. Add Flight
2. Delete Flight
3. Update Flight
4. Helpdesk
5. Return to Main Menu

Enter index of chosen option 2

                                        DELETE FLIGHT
                                        -------------

Enter Flight Number of Flight to be Deleted F030

Displaying Flight Details Before Deletion
('F030', 'Bangalore', 'Hyderabad', '1300 hours', '1500 hours', 25, 25, 800.0, 1500.0)

Record Deleted, Flight number F030 no longer exists
```

ENHANCEMENTS

- The project can be enhanced by using graphics.
- The Project is currently managing airline reservations. It can further incorporate other airline tasks like their personnel management and other ground activities.
- This work can be further beautified, can be made more reliable and more user friendly.
- More validations could have been incorporated.
- More Queries can be added which can extract useful information from the airline database so that relevant data can be retrieved effectively whenever required.

HARDWARE AND SOFTWARE REQUIREMENTS

The hardware and software requirements for Programming in Python are as follows:

- Processors: Intel Atom® processor or Intel® Core™ i3 processor.
- Disk space: 1 GB.
- Operating systems: Windows* 7 or later, macOS, and Linux.
- Python* versions: 3.6.X or later.

This project work has been carried out on Windows 7 Operating System using Python 3.7 for Windows and is compatible with Python 3.7 for Mac OS.

ANALYSIS

- The making of this project was a great learning process for me. Because of this project, I came to know about the online ticketing process of the various reservation systems and got some idea of how softwares are designed for big companies for commercial purposes.
- Both Python and MySQL used in this project are Open Source softwares freely downloadable from the internet, thereby making it easy for us to work with them.
- I learned many new commands in Python, which I never knew and which proved to be of great help while doing programming.
- This also made me realize the power and simplicity of Python as a programming language and the ease with which it can be coupled with MySQL so that we can develop extremely detailed and useful database applications as beginners in a short span of time.
- It also gave me an idea of what exactly programmers do and what challenges they face while designing programs.

BIBLIOGRAPHY

- Computer Science with Python by Sumita Arora (Class XI& XII)
- Python Programming - A Modular Approach, by Taneja & Kumar, Pearson.
- Introduction to Programming using Python by Liang, Y.D., Pearson.
- WEBSITES
    1. www.icbse.com
    2. www.geeksforgeeks.org